A NEW EXACT CLOSEST LATTICE POINT SEARCH ALGORITHM USING LINEAR CONSTRAINTS

Weiyu Xu and Babak Hassibi

Department of EE, California Institute of Technology, Pasadena,CA
E-mail: weiyu,hassibi@systems.caltech.edu

ABSTRACT

The problem of finding the closest lattice point arises in several communications scenarios and is known to be NP-hard. We propose a new closest lattice point search algorithm which utilizes a set of new linear inequality constraints to reduce the search of the closest lattice point to the intersection of a polyhedron and a sphere. This set of linear constraints efficiently leverage the geometric structure of the lattice to reduce considerably the number of points that must be visited. Simulation results verify that this algorithm offers substantial computational savings over standard sphere decoding when the dimension of the problem is large.

Index Terms— closest lattice point search, convex programming, sphere decoder, maximum-likelihood, complexity

1. INTRODUCTION

We address the problem of *exact* maximum likelihood(ML) detection for an integer signal vector that is transmitted through a linear Gaussian vector channel. Namely we want to recover the $m \times 1$ signal vector $\mathbf{x}$ from the $n \times 1$ received noisy vector $\mathbf{y}$,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{H}$ is the known $n \times m$ real-numbered channel matrix and $\mathbf{n}$ is the additive channel vector noise and $\mathbf{x}$ is an integer vector from a alphabet set of cardinality L. Here we assume without loss of generality that the alphabet is the scaled and shifted PAM constellation $D = \{0, 1, ...L-1\}$. (Later, we also consider infinite lattices where the alphabet set is exactly the integer set.) This problem arises in many communications applications such as multiuser detection for CDMA system[5], multiple input multiple output signal detection[1][2], lattice coding[3] and cryptography[4]. Assume that the noise vector consists of independent identically distribute (i.i.d.) Gaussian random variables, the ML detector is given by

$$\underset{x \in D^m \subset Z^m}{\operatorname{argmin}} \ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{2}$$

This work was supported by the Caltech Information Science and Technology Fellowship and Caltech's Lee Center for Advanced Networking.

Clearly the optimal ML detector tries to find the closest lattice point $\mathbf{H}\mathbf{x}$ to the vector $\mathbf{y}$ in a lattice whose basis vectors are the columns of $\mathbf{H}$. ML detection has the merit of minimizing the sequence error probability and achieving the maximum receive diversity, but it is known that the problem of closest lattice point search is NP-hard[5]. One straightforward way to obtain the closest lattice point is the brute-force search, in which the value of (2) is computed for each of the L^m hypothesis. However, the complexity of brute-force search method grows exponentially in m and is not desirable even for a small m. There are efficient convex-programming based algorithms for lattice decoding, such as the semidefinite programming method [18]; however,they do not guarantee optimal performance. Sphere decoder is a popular efficient implementation of exact ML detection which restricts the search of the closest lattice point to a sphere [6][7][11]. Although it has recently been shown that over a wide range of dimensions and SNRs, the sphere decoder can be used to find the exact solution with an expected complexity that is roughly cubic in the dimension of the problem,namely m[13], when the SNR is too low and/or if the dimension of the problem is too large, the complexity of the sphere decoder becomes prohibitive. Actually, under fixed SNR, the expected complexity of the sphere decoder will grow exponentially in m[8]. To overcome the limits of sphere decoding, a branch-and-bound technique was used in [9] to speed up the sphere decoder while still having the exact solution. However, sphere decoders can overlook some geometric structure of the lattices so that it may have much higher complexity compared to other optimal methods in some cases. For example, if the columns of the matrix $\mathbf{H}$ are orthogonal to each other, the matched filter can find the optimal solutions with complexity quadratic in m, but the sphere decoder will eventually end up in exponential complexity with respect to m. This shows that sphere decoder does not fully utilize the geometric information of the lattice, especially when the columns of the channel matrix are near orthogonal.

In fact, in many communications scenarios, the channel matrix has near-orthogonal columns. For example, CDMA systems where orthogonal or near orthogonal signature sequences are used, or precoded MIMO systems when channel state information is available at the transmitter. Even if

the channel matrix columns are not near-orthogonal, lattice-basis reduction methods can be applied to make them near-orthogonal[4]. Of course, the columns of a channel matrix will seldom be exactly orthogonal because of imperfect signature sequences in CDMA systems or imperfect channel feedback for precoded MIMO systems.

Motivated by a set of linear conditions that we find necessary for the maximum-likelihood sequence to be in a branch of the signal space, we propose a convex programming based algorithm which reduces the search complexity for the exact closest lattice point by efficiently taking into account the geometry of the lattice. Simulations suggest that this convex programming based algorithm can be more efficient than sphere decoder.

2. NECESSARY LINEAR CONDITIONS FOR THE CLOSEST LATTICE POINT

In this part, we will derive the set of linear conditions necessary for the closest lattice point of an infinite integer lattice, namely a lattice where any element of $\mathbf{x}$ can take arbitrary values in the integer set Z.

Lemma 1. Suppose that $\mathbf{x}^*$ corresponds to the closest lattice point. Then for any $k \in \{1, 2, ...m\}$, the sequence $\mathbf{x}^*_{(k+1):m} = (x^*_{k+1}, ..., x^*_m)^T$ must be in the polytope

$$\begin{aligned}\Omega &= \{\mathbf{x}^*_{(k+1):m} | -2{\mathbf{x}^*_{(k+1):m}}^T \mathbf{H}^T_{k+1:m}\mathbf{H}_{1:k}(\mathbf{x}_{1:k} - \mathbf{x}^*_{1:k}) \\ &+ (2\mathbf{y} - \mathbf{H}_{1:k}(\mathbf{x}_{1:k} + \mathbf{x}^*_{1:k}))^T\mathbf{H}_{1:k}(\mathbf{x}_{1:k} - \mathbf{x}^*_{1:k}) \leq 0, \\ &\forall \mathbf{x}_{1:k} \in Z^k \text{ such that } \mathbf{x}_{1:k} \neq \mathbf{x}^*_{1:k};\}\end{aligned}$$

Here $\mathbf{H}_{i:j}$ denotes the matrix consisting of the i-th to j-th columns of $\mathbf{H}$ and $\mathbf{x}_{i:j}$ denotes a vector consisting of the the i-th to j-th elements of $\mathbf{x}$.

Proof: Referring to the definition of the closest lattice point,

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}^*\|^2 \leq \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \forall \mathbf{x} \in Z^m \tag{3}$$

If we choose a sequence $\mathbf{x}$ such that its last $m-k$ symbols are the same as those of $\mathbf{x}^*$, then we have

$$\begin{aligned}&\|\mathbf{y} - \mathbf{H}\mathbf{x}^*\|^2 - \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \qquad (4)\\ =\ &(\mathbf{y} - \mathbf{H}\mathbf{x}^* + \mathbf{y} - \mathbf{H}\mathbf{x})^T(\mathbf{H}(\mathbf{x} - \mathbf{x}^*)) \\ =\ &(2\mathbf{y} - \mathbf{H}\mathbf{x}^* - \mathbf{H}\mathbf{x})^T(\mathbf{H}(\mathbf{x} - \mathbf{x}^*)) \\ \leq\ &0.\end{aligned}$$

Since $(\mathbf{H}(\mathbf{x} - \mathbf{x}^*))$ only depends on the first k symbols, the final inequality in (4) is linear in $\mathbf{x}^*_{(k+1):m}$,the last $m - k$ symbols. After manipulation of (4), we can get the linear constraints for polytope Ω.

The number of linear constraints defining the polytope in Lemma 1 is exponential in m. By relaxing the considered polytope, we have a more concise set of m linear constraints as given in Lemma 2.

Lemma 2. Suppose that $\mathbf{x}^*$ leads to the closest lattice point, then $\mathbf{x}^*$ must be in the polytope

$$\Omega' = \{\mathbf{x}^* | \forall i, x^*_i - \frac{1}{2} \leq \frac{{\mathbf{h_i}}^T\mathbf{y} - \sum_{j=1, j\neq i}^m {\mathbf{h_i}}^T\mathbf{h_j}x^*_j}{{\mathbf{h_i}}^T\mathbf{h_i}} \leq x^*_i + \frac{1}{2}\}$$

Proof: We get Lemma 2 from Lemma 1 by confining the $\mathbf{x_{1:k}}^*$ in Lemma 1 to the single symbol ${x_i}^*$. Some algebra work the leads to Lemma 2. Actually, the necessary conditions in Lemma 1 imply that if the interference from the other symbols are canceled, x^*_i will be the nearest constellation point to the matched filter output for symbol x_i.

To the best of our knowledge, the use of the linear inequalities of Lemma 1 and Lemma 2 for the closest lattice point search is new. One exception is [10], where the author proposes similar, yet simplified inequalities for an exact maximum-likelihood decoding algorithm for synchronous CDMA systems. However, the two rules derived in [10] can not fully utilize all possible linear constraints. Although the algorithm there works efficiently for the lightly-to-moderately CDMA system, and is even faster than semidefinite programming based suboptimal algorithms, it will often reduce to a brute-force search algorithm when the columns of the channel matrix have a large crosscorrelation. So there is a need to look for algorithms which can both efficiently in the case of small and large crosscorrelations.

3. CONVEX PROGRAMMING METHOD

In the traditional sphere decoder, we restrict the searched lattice points to a sphere and prune the tree branches out of the sphere. For example, if the sphere decoder sits on a node of the k-th layer of the searching tree, the sphere decoder only keeps the inside-sphere tree nodes of the next layer and search down the tree structure along the corresponding branches. With additional linear constraints in Lemma 1 and 2, we may have smaller feasible sets for subsequent symbols by restricting the search to the intersection of the polytope Ω' and the sphere. So in the new algorithm, we perform the same sequential tree search method as in the sphere decoder[11], but we will shrink the search space by imposing the new linear constraints. Suppose that we want to see whether the ML sequence belongs to the sequence set whose first k symbols are $x^*_1, x^*_2, ..., x^*_k$. By the necessary conditions in Lemma 2, we have m linear inequality constraints for the remaining $m-k$ symbols.Combining the sphere radius constraint as in sphere decoder, we have the following convex program for the feasible lower-bound of x_{k+1}(for the infinite lattice).

$$\begin{aligned}\min\ & x_{k+1} \qquad (5)\\ \text{subject to } & \mathbf{x}_{1:m} \in \Omega' \\ & \mathbf{x}_{1:m} \in B \\ & \mathbf{x}_{1:k} = {\mathbf{x}^*}_{1:k};\end{aligned}$$

Here $B = \{\mathbf{x} | \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \leq R^2\}$, where R is the sphere radius initialized, say, the metric for the zero-forcing decision sequence of $\mathbf{x}$. By solving the convex program in (5), we can have a lower bound for x_{k+1} and similarly, by doing the maximization, rather than minimization, we can also have an upper-bound for x_{k+1}. We then perform tree search between the feasible integer interval to the $k+1$ level as in the standard tree search algorithm. The tree search procedure will be exactly the same as that used in the sphere decoder. Also similar to the sphere decoder[13], for all full-length sequences belonging to the intersection of the sphere and the polyhedron, we compute their corresponding distance and pick the one which is the smallest.
In summary, the algorithm is as follows:

1. Set $\mathbf{x}^* := (0)^T$, $k := 0$, $R := \infty$, $LB_{1:m}$:=0, and $UB_{1:m} := 0$;compute the zero forcing sequence $\mathbf{x}'$ for $\mathbf{x}$ and update R to the corresponding distance metric;

2. Find the lower bound and upper bound for x_{k+1} by solving the convex programming (5). Update LB_{k+1} and UB_{k+1}. If there is no feasible point for x_{k+1}, go to 3; else set $\mathbf{x}^* := (x_1^*, x_2^*, ...x_k^*, LB_{k+1})^T$, go to 4;

3. Find the maximum index $k_1 \leq k$ such that $x_{k_1}^* < UB_{k_1}$. If no such k_1 exists, go to 5; else set $k := k_1$, and set $\mathbf{x}^* := (x_1^*, x_2^*, ...x_{k-1}^*, 1 + x_k^*)^T$, go to 2;

4. If $(k+1) = m$,

   {running x_m^* over all its feasible values to evaluate the distance metrics for x^*. If a better metric is found, replace $\mathbf{x}'$ with the new sequence x^* and update R.
   Find the maximum index $k_1 \leq k$ such that $x_{k_1}^* < UB_{k_1}$. If no such k_1 exists, go to 5; else set $k = k_1$, and set $\mathbf{x}^* = (x_1^*, x_2^*, ...x_{k-1}^*, 1 + x_k^*)^T$, go to 2);}

   else if $(k+1) \neq m$,

   {set $k = k + 1$,Go to 2;}

5. Output $\mathbf{x}'$ as the maximum-likelihood sequence.

Theorem 1: The output of the convex programming based algorithm is the maximum-likelihood sequence that generates the closest lattice point to $\mathbf{y}$.
Proof: Theorem 1 follows from Lemma 2 and the fact that the output of the convex programming based algorithm is the sequence that provides the smallest metric among all the sequences that satisfy the necessary constraints in Lemma 2 and the sphere radius constraint.

For the convex-programming based approach, at the k-th level of the tree, we will have at most $(2m + k)$ linear constraints, one convex quadratic constraint and m variables. It is well known that there exists efficient algorithms for solving this convex programming which has polynomial-time computational complexity with respect to the number of variables and constraints. Such algorithms include efficient primal-dual interior point methods for convex programming[16]. When the dimension of the closest lattice point search goes large, the dominating factor of the computational complexity will be the number of tree nodes that the algorithm visits [8]. So in some parts of our analysis and simulations, we use the average number of visited tree nodes as a measure of the average computational complexity of the proposed algorithms and sphere decoders. Now we will give a lemma about the computational complexity of the convex programming based new approach.
Lemma 4: If the columns of $\mathbf{H}$ are orthogonal to each other and there is only one closest lattice point to $\mathbf{y}$, then the convex programming based new approach will only visit one node at each level.
Proof: It is straightforward to notice that for any finite R, there is only one constellation point from each layer lying in the polytope Ω'. So the convex programming based approach will only keep one tree node at each level, which results in a polynomial-time algorithm for optimal decoding in this case.

For finite memory communications channel, we will have similar results as in Lemma 4. This shows that the new algorithm can smartly adapt to the geometric structure of the lattices when compared to the traditional sphere decoder.

We further remark that ideas for the infinite lattices can be applied to finite lattices in a similar way. These new linear constraints can also be integrated into the branch-and-bound sphere decoder [9] in a straightforward way, which results in enhanced branch-and-bound sphere decoder. This enhanced branch-and-bound sphere decoder can be regarded as one special case of the convex optimization based algorithm described in this section. We compare the experimental complexity of the enhanced branch-and-bound sphere decoder to other sphere decoders in the next simulation section.

4. SIMULATION RESULTS

In this part, we give some simulation results for the convex programming based algorithm. First, we compare the new algorithm with the standard sphere decoder[13] in terms of the average CPU time used to finish one instance of the exact closest lattice point search for infinite lattices of different dimensions. For solving the convex programming, we use the software package cvx [17] under MATLAB on a 1.73G HZ PC. The lattice generator matrix is square and is generated by adding an identity matrix to a square perturbation matrix whose entries are i.i.d Gaussian random variable with variance 10^{-6}. The entries of additive Gaussian noise are set to have a variance of 1. As we can see, the new convex programming based method can automatically recognize the near-orthogonal structure in the closest lattice point search problem and surpasses the traditional sphere decoder [13] in terms of speed even though the new method needs more com-

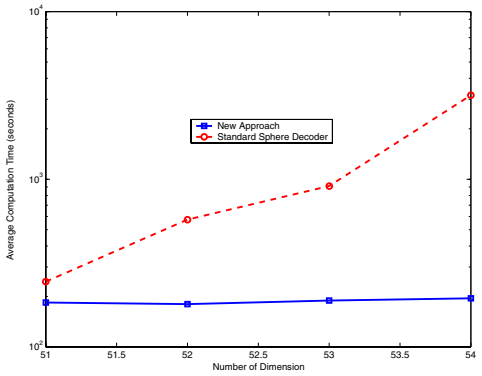

Fig. 1. CPU Time of Two Methods for Closet Lattice Point Search

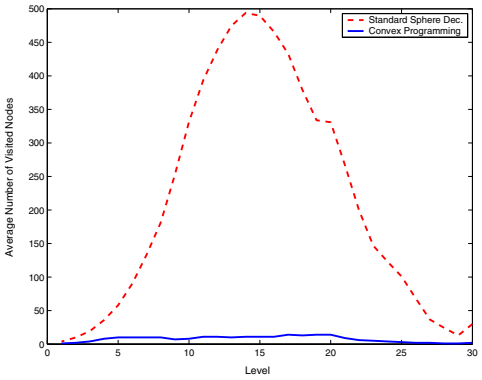

Fig. 2. Comparison of Average Number of Visited Nodes

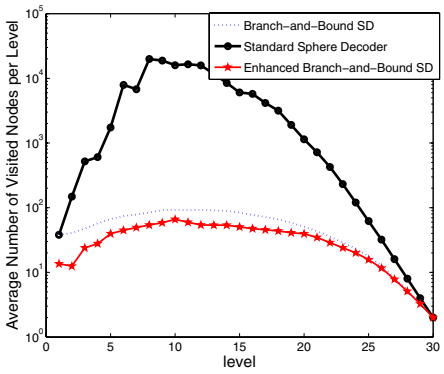

Fig. 3. Comparison of the Average Number of visited nodes

putation for each tree node. One may argue that in this case, zero-forcing decoding is another good candidate, but we are interested in the *exact* closest lattice point search, which is sometimes important for applications in cryptography and computer science [4]. The example we give here is for illustrative purpose. In fact, the new algorithms also works efficiently when the matrix columns are well coupled, which will be shown in Figure 3.

Figure 2 shows the expected number of visited nodes at each level in the closest lattice point search for an infinite lattice, whose generator matrix is a 250×30 random matrix with each entry as a unit-variance Gaussian random variable. The new algorithm considerably reduces the number of tree nodes at each level when compared with traditional sphere decoder.

We now look at a finite lattice with $m = 30, L = 2, SNR = 0dB, n = 30$ and the entries of the lattice generator matrix are i.i.d unit-variance Gaussian random variables. We remark that in this situation, the columns of lattice generator matrices are well-coupled. In Figure 3, we compare the average number of visited tree nodes of enhanced branch-and-bound sphere decoder (using the new linear constraints in this paper), the polytope relaxation based branch-and-bound sphere decoder[9] and the standard sphere decoding algorithm. It can be seen that the newly proposed method outperforms the polytope relaxation based branch-and-bound sphere decoder by a factor of $1.5 \sim 2$ in terms of the average number of tree nodes visited.

5. CONCLUSIONS AND FURTHER DISCUSSIONS

We propose to use a new set of necessary linear constraints in designing efficient and *exact* closest lattice point search algorithms,which has potential applications in communications and other areas. The proposed algorithm obtains the exact solution while greatly reducing the average number of visited nodes by restricting the lattice search to the intersection of a polyhedron and a search sphere. It is especially desirable when the columns of the channel matrix are near-orthogonal. The results in this paper suggest that through analyzing the geometry of the closest lattice point search problem, we may further reduce the average-case computational complexity. The efficiency of optimally or approximately solving the closest lattice point problem through other geometrical approaches have been illustrated in other works [14] and [15]. The new algorithm in this paper can be possibly improved in various aspects. For example, it may be worthwhile investigating whether we can reduce the complexity of convex programming by utilizing the fact that the convex programming problems involved share many common constraints.

6. REFERENCES

[1] E.Telatar, "Capacity of Multi-Antenna Gaussian Channels," AT&T-Bell Labs,Tech.Rep.,1995

[2] H.El Gamal,G.Caire and M.O.Damen,"Lattice coding and decoding achieve the optimal diversity-multiplexing tradeoff of MIMO Channels," *IEEE Tans.Inform. Theory*, pp. 968–985, June 2004;

[3] O.Shalvi, N.Sommer and M.Feder"Signal Codes," *Pro-*

ceedings of Information Theory Workshop, pp. 332–336, 2003;

[4] Daniele. Micciancio,Shafi Goldwasser, *The Complexity of Lattices-A Cryptographic Perspective*,Kluwar International Series in Engineering and Computer Science,2002;

[5] S.Verdu, "Computational Complexity of Optimum Multiuser Detection," *Algorithma*, vol. 4, pp. 303–312, 1989;

[6] E.Agrell,T.Ericsson,A.Vardy, and K.Zeger,"Cloest Point Seach in Lattices," *IEEE Transactions on Information Theory*, vol. 48, pp. 2201–2214, August, 2002;

[7] M.O.Damen,H.E.Gamal,and G.Caire, "On maximum-likelihood detection and the search for the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, pp. 2389–2401, Oct. 2003;

[8] J.Jalden,B.Ottersten,"On the Complexity of Sphere Decoding in Digitial Communications," *IEEE Transactions on Signal Processing*, vol. 53, pp. 1474–1484, April,2005;

[9] M.Stojnic, H.Vikalo, B.Hassibi,"A Branch-and-Bound Approach to Speed Up Sphere Decoder," *Proceedings of Acoustics,Speech and Signal Processing*, vol. 3, pp. 429–432, 2005;

[10] Fatih Algoz, "A New Algorithm for Optimum Multiuser Detection in Synchronous CDMA Systems," *AEU International Journal of Electronics and Communications of Engineering*, vol. 57, pp. 263–270,2003;

[11] A.D.Murugan, H.El.Gamal, M.O.Damen,and G.Caire, "A Unified Framework for Tree Search Decoding:Rediscovering Sequential Decoder", to appear in *IEEE Transactions on Information Theory*, 2006;

[12] U.Fincke and M.Pohst, "Improved Methods for Calculating vectors of short length in lattice,including a complexity analysis," *Mathematics of Computation*, vol. 44,no. 170, pp. 463–471,April,1985;

[13] B. Hassibi,H.Vikalo, "On the Sphere Decoding Algorithm. I. Expected Complexity" *IEEE Transactions on Signal Processing*, vol. 53, pp. 2806–2818,2005;

[14] H.Artes,D.Seethaler and F.Hlawatsch, "Efficient detection algorithms for MIMO Channels:A Geometrical Approach to approximate ML detection," *IEEE Transactions on Signal Processing*, November,2003;

[15] I.Motedayen,A.Anastaspopoulos, "Optimal Joint Detection/Estimation in Fading Channels with Polynomial Complexity," *IEEE Transactions on Information Theory*,submitted;

[16] S. Boyd and L. Vandenberghe,*Convex Optimization*,Cambridge University Press, 2004.

[17] Michael Grant, Stephen Boyd,and Yinyu Ye, *CVX: Matlab Software for Disciplined Convex Programming*,http://www.stanford.edu/ boyd/cvx/

[18] Amin Mobasher, Mahmoud Taherzadeh, Renata Sotirov, and Amir K. Khandani, A Near Maximum Likelihood Decoding Algorithm for MIMO Systems Based on Semi-Definite Programming, Submitted to IEEE Transactions on Information Theory, Aug. 2005